

Methods to produce flexible and accessible learning resources in mathematics: overview document

Emma Cliffe

May 2012

Introduction

This work was carried out as part of a HESTEM Mathematics Curriculum Innovation project. We have explored methods to produce flexible and accessible learning resources for mathematics with a focus on producing a guide for staff in HEI to enable them as individuals to create flexible resources efficiently and robustly.

This document provides an overview of:

- The literature identifying the required formats and technologies available for transforming or creating mathematical documents.
- The data collected from staff on their current document production methods.
- The process we went through to produce the methods.
- The remaining costs, risks and barriers.

The methods themselves are not documented here. Please refer to the other project documents.

Contents

1 Overview of the literature	1
2 Staff and student input	3
2.1 Student view	3
2.2 Staff view	6
2.2.1 Teaching staff	6
2.2.2 Support staff	9
3 Production of the methods	11
4 Evaluation	12
5 Costs, risks, barriers	17
5.1 Further development and sustainability	26
Bibliography	26
A Technologies list	30

1 Overview of the literature

A review of the literature and available technologies confirmed the formats that we might need to produce but also furnished us with extensive lists of technologies which might assist us. We present an overview of this literature here.

The provision of partial notes in mathematics courses is perceived by students as beneficial to their learning and is strongly related to high academic performance [1]. However, a significant barrier for disabled students is the suitability of the learning resource format provided. Many of the contributions to the Good Practice on Inclusive Curricula in the Mathematical Sciences guide [2] highlight the need for full notes in specific formats to be provided prior to classes. Some case studies available in mathematical subject areas such as on the Strategies for Creating Inclusive Programmes of Study [3] and Disabilities: Academic Resource Tool [4] also confirm the need for full notes, sometimes in specific formats.

Some students require full notes in Braille [6, 5, 2], possibly using a Braille display [7, 2] either for Braille mathematics or direct access to the \LaTeX code [8]. Other students require large print and authors highlight that large print is not simply a matter of using a larger font but also requires some or all of changes to spacing, page layout, layout of the mathematics, font or colour [10, 8]. However, students (and staff!) are likely to have difficulty creating a large print version even if provided with the \LaTeX source [9], the lack of line-breaking in the equations being a primary issue and the MathType format is highlighted as being useful [2]. The RNIB Clear Print guidelines are a useful starting point [11] and note the minimum font size etc. considered advisable for general resources. For students reading in Braille or large print access to full notes in class act as an alternative to the board and permit the student to follow the lecture content.

Students who are d/Deaf are not able to lipread, watch a BSL/English interpreter or lipspeaker and take notes. If a student does not have some form of note-taker they will not be able to write down what you write on the board. For a student lipreading all verbal information given while facing the board e.g. to annotate working, will be lost [12]. Some practices known to be beneficial in d/Deaf education are available to lecturers including the provision of visual organisers, using a collaborative, case study, problem-solving approach (where possible) and pre-teaching (or, at least, enabling preparation) of specialised vocabulary [13]. This suggests that notes which clearly highlight terminology, use visual

organisers and include examples are likely to be helpful.

Some disabled students experience difficulties copying precisely, knowing what to write down, maintaining concentration or their place in text or retaining definitions in memory. This might include some students with specific learning difficulties (e.g. dyslexia, dyspraxia), students with Asperger syndrome, mental health and fatigue conditions. Access to full notes in a suitable format enables the main activity in class to be the desired engagement with concepts and logical arguments. Formats including those presented in sans serif fonts, clearly structured documents, coloured backgrounds, formats which can be adapted by the students to their font, spacing and colour requirements, formats with colour or structural highlighting of equation scope, audio formats or formats which can be read loud, video formats of real time manipulation, visual organisers such as mind-maps and flow diagrams, notes which can be annotated in class and formats from which formula can be copied and pasted or notes electronically annotated are all reported as being of assistance [2, 14, 15, 16, 17, 18].

In summary, the use of visual organisers helps some students as does the provision of editable formats. Other students might require quite specific formats which may include clear print, large print and formats accessible by text-to-speech (e.g. literacy support software used by dyslexic students), screenreader or Braille technologies. Word or PDF documents containing mathematics cannot easily be produced in these formats.

In order to provide such a range of formats we require a method to produce a single master version, which can be updated over time, from which the multiple required formats can be produced automatically. Using a single master is not a new idea [19, 20, 21] but we require viable methods which specifically produce accessible formats.

Cooper [22, 23, 2] neatly captures the technical challenges and general approaches which we might need to take. The use of MathML [24] is a key technology as this enables speech to be produced, equations to scale with the surrounding text and reflow, linebreaking as necessary. Scalable graphics can also be used to permit scaling of equations [25] and automated linebreaking is possible in \LaTeX [26].

MathML is not designed for humans to read and write directly! The World Wide Web Consortium (W3C) maintains extensive lists of technologies which produce MathML or convert between MathML and other formats [27] and these formed

a starting point. However, guidance from the literature [28, 29, 30, 31, 32, 33, 34, 35, 36] greatly facilitates comparisons between and understanding of these technologies!

MathML is not the only format we considered. The TeX User Group (TUG) maintains lists of converters between \LaTeX and word processor formats [37, 38] and the American Mathematical Society (AMS) maintain a list of TeX related resources [39].

Finally, the many reference documents on \LaTeX hosted by the Comprehensive TeX Archive Network [40] and the bug trackers of all the software we worked with sometimes gave insights as to the likely cause of misbehaviours of the transform technologies in the face of the author's freedom to use and abuse \LaTeX [41]!

We list the main technologies that were used during the evaluation in Appendix A though some which we incidentally interacted with (mainly other \LaTeX packages) are not listed.

2 Staff and student input

2.1 Student view

Disabled students in mathematics had previously provided input on their requirements, their use of assistive technology and given feedback on the notes already in production. We invited further input for this project and one student updated us on how his use of notes had changed over time.

Lecturers either provide handwritten notes or \LaTeX which has normally been encoded in 10pt, 11pt or 12pt default spacing and fonts.

Formats we have created:

- Clear print: A4, single sided, 12pt helvetica font for English, extra white between paragraphs, extra spacing in maths. Extra space is to enable annotation.

Are provided each year to students who are dyslexic, dyspraxic, have Asperger syndrome, mental health difficulties, conditions causing fatigue and to some students with mobility difficulties based on the recommendation of the DSA Needs Assessor, Disability Adviser or Study Skills Tutor. Notes are also provided to students with temporary conditions at the request of the Director of Studies.

- Why they are needed: “My eyes don’t especially like the font [in the standard course notes] and it is all a bit close together.”
 - Notes are valued for their reading clarity: “Clear PDFs — clarity and portability”; by portability this student is referring to his practice of using an e-book reader to enable him to transport his notes with greater ease.
 - Notes need to be up to date: students lose track in lectures and will quickly get in touch if they are not.
- Large print: A4, single sided, helvetica font for English, extra white space between paragraphs and in mathematics, enlarged diagrams and:
 - 14pt text
 - 17pt text
 - 20pt bold English but not maths, underlining for emphasis, double spaced text with a clear typewriter font for verbatim text
 - 26pt bold English and maths, double spaced, no emphasis

The latter two formats have each been provided to one student only. 14pt and 17pt have been provided on several occasions.

Why they are needed:

- “I could really do with an enlarged font version.”
- “Without the lecture notes there is no point in being there...”

Why they **must** be updated:

- “the confusing part is when lecturers change the order of things round completely”
- “the lecturer often strays from the official course notes.”
- Students receiving these notes can lose track in lectures quite rapidly as they are not able to see the board clearly/at all.

- “Could I have the \LaTeX ?!”: We have provided the \LaTeX sources to two students
 - For one student the comments had to be removed. The student learnt \LaTeX in third year because they could no longer effectively produce handwritten notes and wished to continue annotating the full notes provided.
 - For a second student “human readable” \LaTeX was provided. The screenreader used by the student (or any other available) cannot read the mathematics in PDF or Word documents. The student read the \LaTeX source but found commands present solely for visual layout etc. to impede reading so these need to be removed.

- Editable formats for students who don’t know \LaTeX : requested but can’t be provided...

Why are they needed: because the students often prefer to adjust the materials to their own requirements and are best placed to do so, if we can enable this.

- “I should be okay with [the notes already available from the lecturer] as long as I can adapt them as needed...”
- “I wondered if I could get some editable ones as theres lots of gaps and font size changes on the PDFs online which make it really hard when I’m trying to read them.”
- A regular request: “Can you correct the error on ***** notes?”. Some students find it difficult to ignore or work round errors in notes and genuinely require a correct version but the format means they cannot alter it themselves. This sometimes results in confusing and time-consuming conversations between the student, the staff providing notes and the lecturer: “where exactly is the error?!”.
 - Formats which work with text-to-speech: not requested but students report that... “This doesn’t work...”
 - Some students report their frustration that text-to-speech does not work or ask if it is possible. Since equations are interspersed with text this even impedes students who primarily wish to hear the English text aloud rather than the equations themselves.

2.2 Staff view

Whether staff and departments are willing to use a method to create learning resources is a key concern. To increase the uptake of any proposed methods it is ideal if they reflect those already in use and provide much of the same functionality. We sought input from teaching staff but also from staff employed in producing accessible versions of mathematical documents.

2.2.1 Teaching staff

A survey was directed at teaching staff in mathematics at three institutions to ascertain the type of mathematical resource they typically produced, the underlying format in which they worked (and their minimum requirements for this), the formats in which they provided resources to students and their experience, if any, of supplying accessible notes.

We additionally asked if staff would be willing to share representative samples of their resources with us for research purposes and, with their permission to be anonymously quoted (fully or partially) in any resultant guidance.

We received 45 responses 16 of which also provided representative samples; 12 of the sample sets are available for us to quote from and we hope to provide guidance on working with legacy documents in the future.

Staff produced a wide range of mathematical resources (figure 1). The majority of staff are using a mixture of handwritten, \LaTeX and Microsoft Office formats in which to encode their resources (figure 2).

Staff used a very wide range of methods to create images (there was little duplication) and so there appears to be the need to cater for the inclusion or conversion of multiple image formats. This includes catering, if possible, for images created within using \LaTeX using the `picture` environment and packages such as `xypic`, `pb-diagram` and `pstricks`.

A variety of \LaTeX packages were in use for learning resource production. Respondents listed the following packages (group loosely by purpose) as being packages without which they would not be able to produce resources:

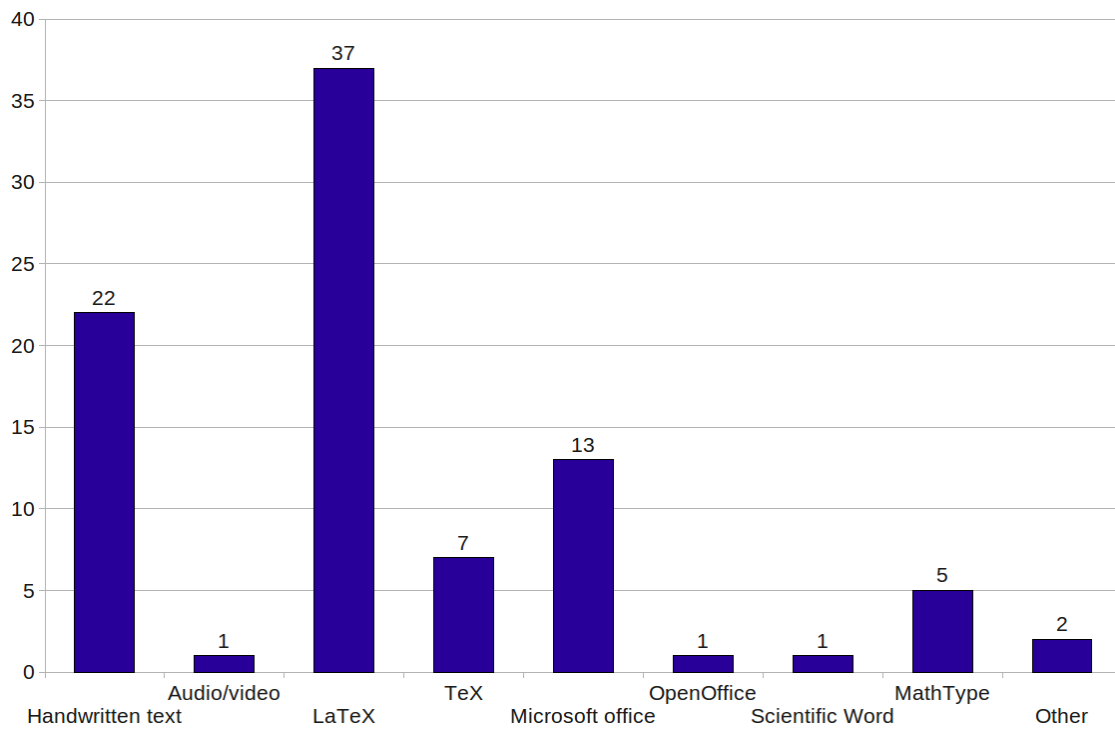


Figure 1: The type of mathematical resources respondents produce.

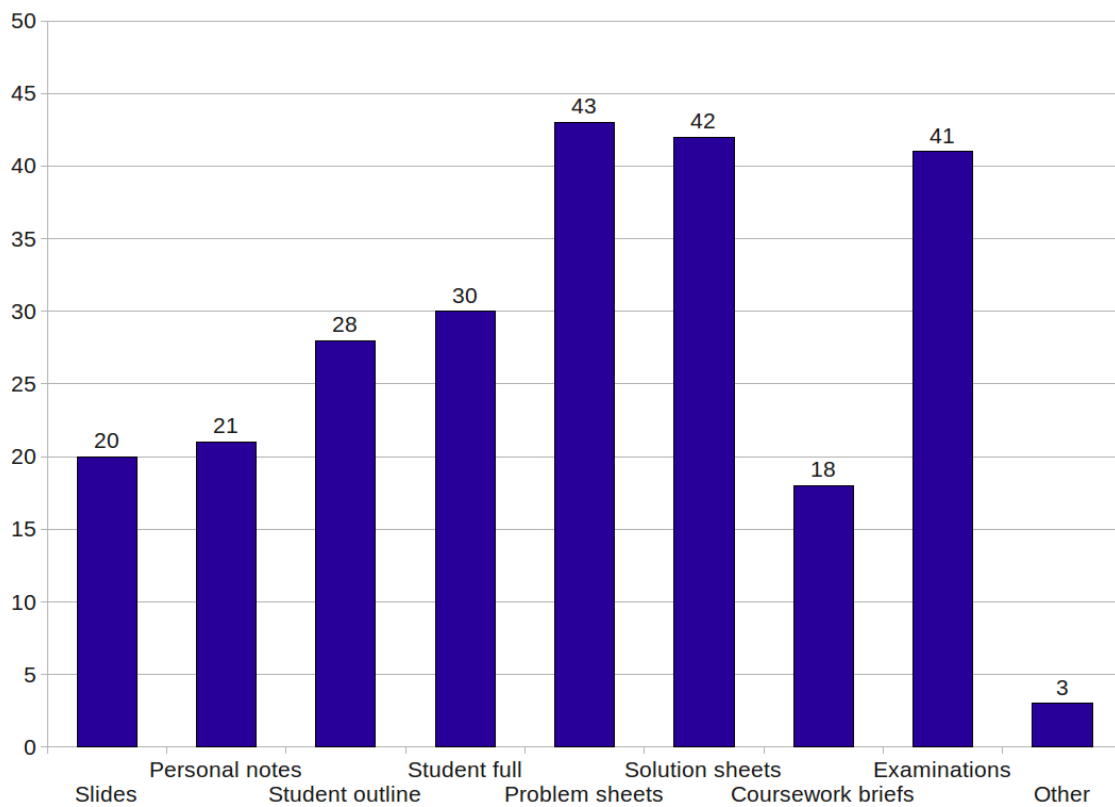


Figure 2: The format in which staff produce mathematical resources

amstex, amsmath, amssymb, amsfonts, amsthm, amsbsy, eucal, mathtools,
 beamer (class),
 graphics, graphicx, epstopdf, xy, epsf,
 color,
 enumerate, fancyhdr, listings, natbib,
 ifthen,
 latexsym, stmaryrd, bm, esint, xspace,
 fontenc

One respondent has his own packages and class files.

We were able to view the preambles of the sample documents and the above does not capture all the packages actually **used**. The exceptions were (in no useful order at all):

epsfig, verbatim, url, paralist, array, calrsfs, mathrsfs,
 psfrag, lscape, pstricks, cancel, cases, geometry, keyval,
 multicol, babel, inputenc, times, pgfpages, newlfont,
 datettime, makeidx, ulem, mathdots, hyperref

Staff tend to provide students with handwritten or PDF documents (figure 3), this latter appears to include resources which were created in Word. One sample set of resources did include Word documents and exported PDF versions.

In response to the question “Have your mathematical learning resources been converted into/produced in any output formats specifically for a disabled student by you or your department?”, 15 respondents said yes, 9 were unsure and the rest had not. Only 1 respondent was able to create the resource format required automatically from their usual production format.

2.2.2 Support staff

A small group of ‘expert users’ employed in producing accessible mathematical documents at two institutions were identified. The ‘expert users’ could not produce the required output formats from a single master though each could produce some of the specific formats to requirements. This included various large and clear print formats in PDF and hard-copy, provision of raw \LaTeX or

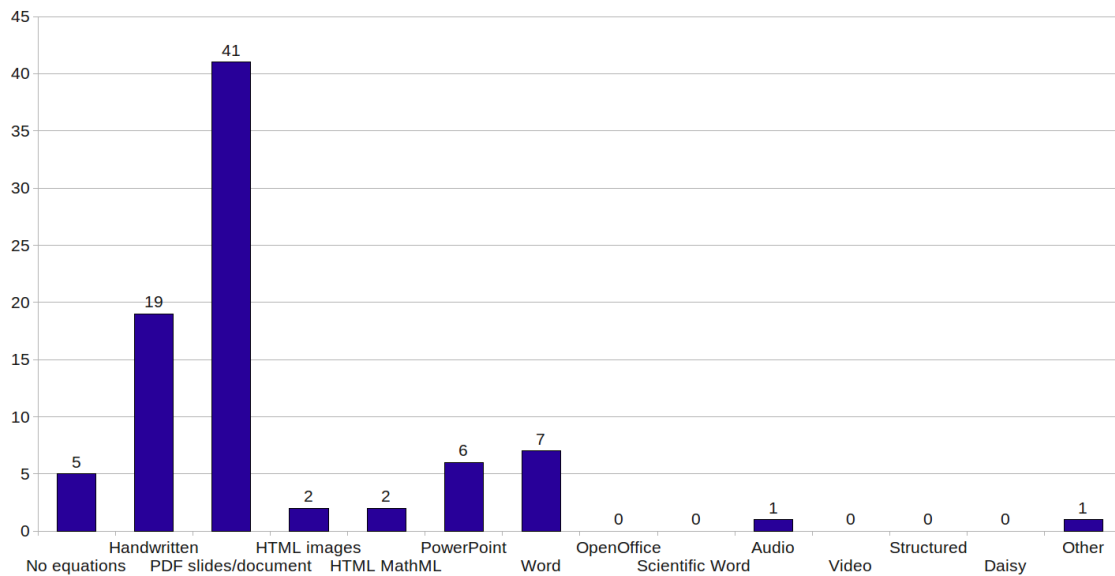


Figure 3: The format in which staff provide mathematical resources to students

'human readable' \LaTeX directly to students and MathML enabled formats to permit screenreader and text-to-speech access, created using on-the-fly translation from \LaTeX snippets in web pages (using LaTeXMathML, MathJax and ASCIIMathML). Staff were also aware of full document transformation and the benefits of MathType but notes were not produced using Word locally.

The cost of producing inflexible learning resources was noted. These require costly adjustments to be made in reaction to individual requirements and this may be the work of specialist support staff. In the Department of Mathematical Sciences at the University of Bath, over the last seven years, support staff have produced full notes of courses in a variety of accessible formats.

Staff found it difficult to reuse old versions of resources though sometimes they found it possible to produce a single version which was acceptable to all the students requiring notes on a specific module. However, this was not always the case, new versions of resources were sometimes produced for each specific set of requirements. Staff found that the versions they created for particular students are not updated by teaching staff. As a result new formats for some modules were produced three times in the seven year period.

There were examples where students accepted a format that was not completely to their requirements because a better format could not be created e.g.

Speech but no useful Braille output, non-editable format.

3 Process used to the produce the methods

We used a set of documents each capturing a single component of structure e.g. a list, an equation array, an image or a table. The equations used were chosen to cover a range of one and two dimensional layouts e.g. symbols mixed with relations, sub- and super-scripts, matrices, roots or fractions. We transformed each document with each technology and evaluated the results.

We next used Representative sample documents provided by survey respondents as complex inputs to test the identified transform technologies. Often no output was produced; the cause of this was recorded as a constraint and removed to produce a new input document. If output could be produced this was evaluated. We discarded document transform technologies if they were too constrictive or unstable for our purposes or if the results difficult to realistically evaluate. In some cases we identified problems which we felt could be overcome. These were recorded as barriers but proof of concept work rounds were produced and evaluated.

Some of these experiences are recorded in section 5. A full account of the decisions made is outside the scope of an overview document!

We produced integrated test documents formed of structural components which met the recorded constraints then extended these to include a range of mathematical symbols and commands (those available in \LaTeX with limited packages on the one hand and in Word on the other). The remaining transform methods were applied simultaneously to these and each output evaluated. We iteratively adjusted the symbol, command, style and working process constraints until all output formats were correct within certain tolerances. These tolerances were recorded as remaining barriers.

Finally, the constraints and methods were documented and a member of staff not involved in the above trialled the process.

4 Evaluation

Evaluation had two objectives: to measure the progress in attaining correct output documents and to evaluate the impact of using the final method in document production. The iterative evaluation and adjustment of the transformation methods, first at a component level and then at an integrated document level, using an 'expert user' to carry out all evaluations, is akin to software testing methodologies in which access to users is limited. The evaluation goal was to locate and record defects in the output (or, if no output was produced to determine the cause). Output was checked visually and with relevant assistive software, for correctness by comparison and for usability via a heuristic inspection.

The baseline for document production was established from staff input. None of the sample documents could be used to automatically produce multiple formats and most could not be transformed successfully without alteration. Of the 45 survey respondents 15 had had resources produced in an output format specifically for a disabled student and a further 9 were unsure whether this had occurred. Only 1 member of staff reported that they had been able to produce the required format without assistance.

To evaluate the impact of using the methods we asked an expert user to encode a series of handwritten sample documents. We evaluated the outputs produced and the expert user answered a series of questions to ascertain their views on the methods, the additional costs involved and the challenges experienced.

The expert user took one hour to comprehend the main restrictions of the process but frequently needed to search the guidance documents for details. On the first attempt details were missed which caused all diagrams to be lost. Her feedback also exposed an incomplete understanding of the restrictions. Her difficulties highlighted the exacting technical nature of the task even for an expert user. The searches for technical information imposed an extra cost to producing the documents. She felt she would eventually learn the restrictions and, asked to disregard the overhead, stated "I believe this is as long as it would have taken me working in the usual way [to create one specific format]". On reaching full understanding of the methods, she concluded they were flexible enough and saw they could be expanded in the future. She noted that "establishing a common standard for producing notes is the best way to not only produce accessible notes but to ensure easy transition between lecturers for a course". The main costs beyond production overheads were software set up on two operating sys-

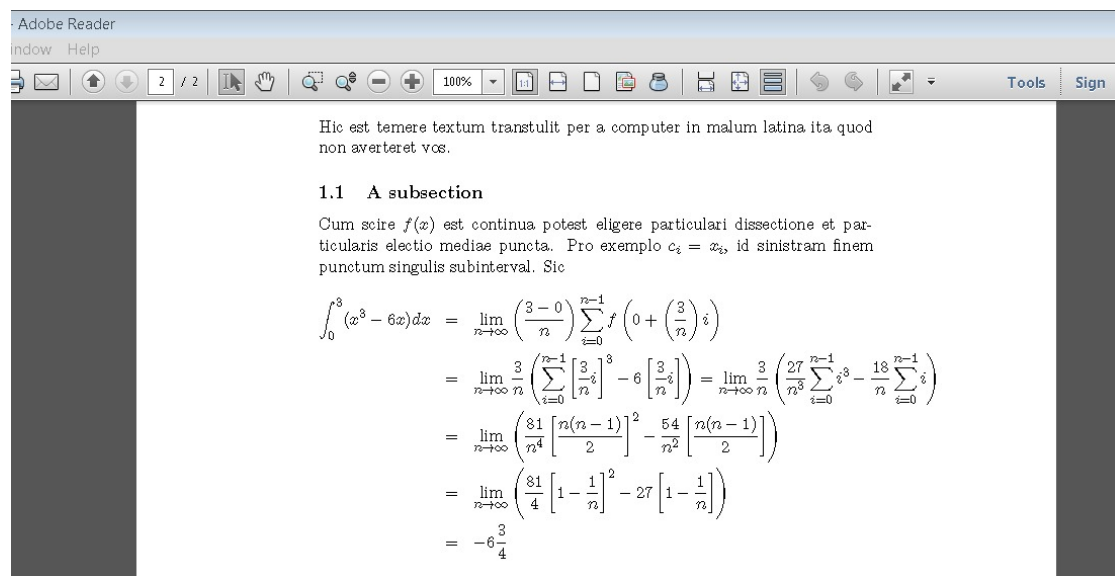
tems and performing the transformations for each document. These overheads are not insubstantial and again required instructions to be followed precisely.

The experience of the expert user and the integrated document evaluations confirmed that while production of correct documents is possible the methods are technically exacting (even the smallest deviations can lead to no or incorrect output). The required precision is due to missing or incorrect software functionality. Staff specialising in accessible mathematical document preparation have experience in following precise procedures in this work. We concluded that they will be best placed to produce the master documents in the first instance, to transform the documents as required for individual students and to advise teaching staff updating masters. The final stage of the evaluation would have involved user tests by teaching staff. The conclusions already drawn suggested that this was unrealistic. It was inappropriate to ask staff to spend a substantial amount of time attempting to test the methods so the plan was abandoned.

We have derived and documented structure, command, style and symbol constraints which permit multiple formats to be produced. Staff specialising in mathematical document production for disabled students will be able to produce, from a single master (figure 4), formats including large print PDF (figure 5), also suitable for small screens such as some e-book readers, browser independent formats Figure 3: text-to-speech enabled format suitable for small screens and large print produced from \LaTeX master on screen, formats accessible to text-to-speech technologies (figure 6), editable formats (figure 7) and formats suitable for screenreader use (speech and/or some form of Braille).

The project has positively impacted practice by:

- increasing the knowledge and expertise of staff who produce and advise on the production of accessible learning materials in mathematical sciences and
- providing a method to reduce resource production and maintenance costs, helping to ensure sustainable access for disabled students to formats suited to their needs.

Figure 4: standard PDF output from \LaTeX

- Adobe Reader
dow Help

2 / 3 100% Tools Sign

dissectione et particularis electio mediae puncta. Pro
exemplo $c_i = x_i$, id sinistram finem punctum singulis
subinterval. Sic

$$\begin{aligned} \int_0^3 (x^3 - 6x)dx &= \lim_{n \rightarrow \infty} \left(\frac{3-0}{n} \right) \sum_{i=0}^{n-1} f \left(0 + \left(\frac{3}{n} \right) i \right) \\ &= \lim_{n \rightarrow \infty} \frac{3}{n} \left(\sum_{i=0}^{n-1} \left[\frac{3i}{n} \right]^3 - 6 \left[\frac{3i}{n} \right] \right) \\ &= \lim_{n \rightarrow \infty} \frac{3}{n} \left(\frac{27}{n^3} \sum_{i=0}^{n-1} i^3 - \frac{18}{n} \sum_{i=0}^{n-1} i \right) \\ &= \lim_{n \rightarrow \infty} \left(\frac{81}{n^4} \left[\frac{n(n-1)}{2} \right]^2 - \frac{54}{n^2} \left[\frac{n(n-1)}{2} \right] \right) \end{aligned}$$

2

Figure 5: large print (20pt, reflowed mathematics) produced from L^AT_EX master

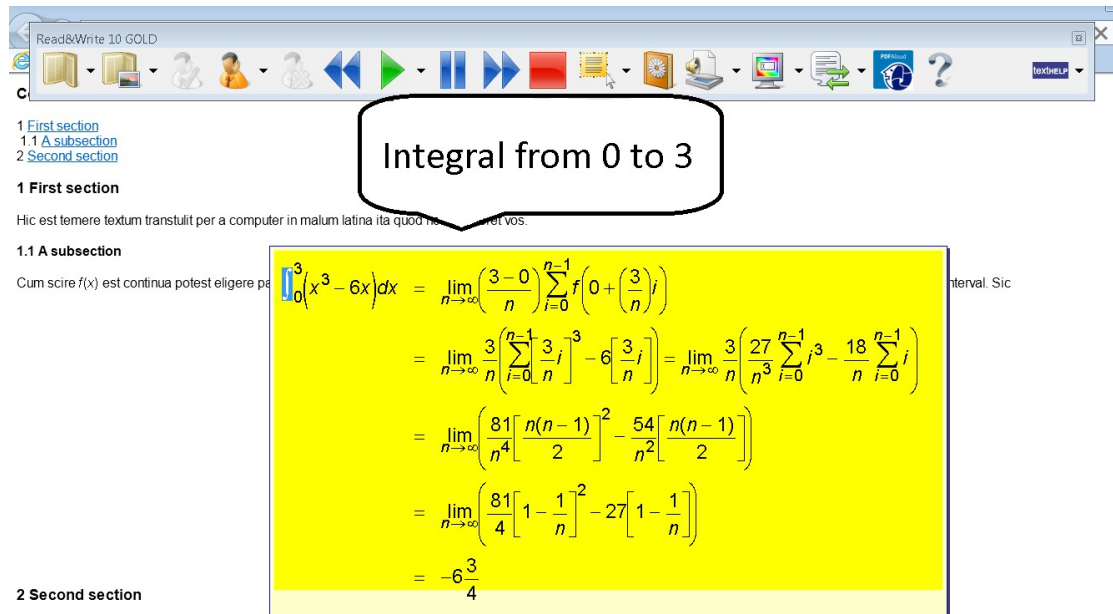


Figure 6: text-to-speech enabled format produced from \LaTeX master

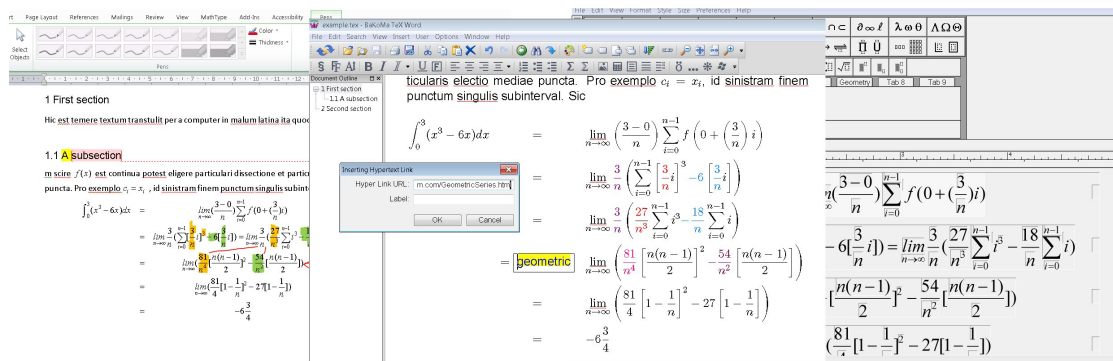


Figure 7: editable formats produced from \LaTeX master (with example annotations and alterations in progress)

5 Costs, risks and remaining barriers

It is **strongly** recommended that the reader does not attempt to read this section if they have not at least skimmed the guidance on document production and transform methods from the main site. It is questionable how much sense the section will make without some insight as to the restrictions and requirements of the methods.

We recommend that separate **Linux and Windows 7 machines** be used (the Linux software requires only command line interface). We used a dual boot machine and spent quite a lot of time rebooting...

The following is a list of necessary software on the two operating systems. Please see appendix A for links to further information about the software.

Linux:

- TeXLive
- ImageMagick
- C, Python and Flex.
- Stix fonts or other fonts to enable a visual check of MathML output in Linux.
- A PDF viewer and web browser.
- TeX4ht.
- PlasTeX compiled from source with our changed files incorporated — see the transformations document for instructions.
- Our proof of concept code for producing \LaTeX for large print PDF and for input to TeX4ht in order to produce LibreOffice documents. Requires Flex and C to compile — see the transformations document for instructions.

None of the above software has a direct cost however, the installation and maintenance of the system will require **staff time**.

Windows:

- Software which must be **purchased**:
 - Office 2010.
 - MathType (minimum version 6.8 required).
 - Word2TeX.
 - Software for testing purposes only:
 - * Multiple browsers.
 - * Screenreader and literacy support software from the list: http://www.dessci.com/en/solutions/access/atssupport.htm#Reader_Tools
 - * BaKoMa.
 - * ChattyInfty (if exploring options for speech and Braille).
- Software which has no direct cost but will require **staff time** to install and maintain:
 - LibreOffice.
 - MathPlayer (version 2.2 required).
 - \LaTeX -access and LaTeXLex (if exploring options for speech and Braille).

Time for an expert user to **gain a basic understanding** of the key restrictions is approximately 1 hour but from experience they will need to produce and transform **several practice documents** to reach a full understanding. Once this has occurred the only remaining time overhead in the creation of the master documents is the **extra work required to incorporate EPS diagrams into documents**.

Transformation of the documents cannot be truly automated as a single transform may rely on two operating systems and on the use of software with a GUI interface. The **transformations are completed without change to the master document**. With careful use of make files, scripts, macros and storage of files **many steps can be completed with limited user intervention**. **Some steps appear to scale badly** with the number of equations in a document but we have not quantified this.

We recommend that support staff focused on the task produce masters, transform from masters to other formats as required and advise staff updating master documents. **Specialist staff time is likely to be the most substantial cost**. Assuming such staff are already employed to produce accessible formats this cost will diminish over time provided that master documents are updated.

Use of open source, free software

All transforms from \LaTeX rely on open source, free software. There remains the risk that this software is not updated in the future and ceases to be viable on a modern computer. This is a wider question for the mathematical community which uses these tools. To attempt to quantify this risk we provide **factual** information about some of the tools, as of 31st May 2012, which may allow a reader to investigate further.

- TeX4ht: was created by Eitan Gurari at Ohio State University, he died unexpectedly in June 2009. It is stated on the new TeX4ht website, hosted by TUG (TeX User Group) that “With the encouragement and support of Eitan’s family, CV Radhakrishnan and Karl Berry are now working on the package. Involvement by other volunteers, from bug reports to major new development, is welcome and needed”; “No full post-Eitan release has been made to date”; “In TeX Live, we have installed small updates to hyperref.4ht, biblatex.4ht, and nameref.4ht (and no other files). All other development changes remain solely in the source repository.”. The source repository was last updated on 25th April 2012.
- PlasTeX: Kevin Smith is the contact name, registered on sourceforge in October 2004, he last updated the source on 30th April 2012. The sourceforge site notes that it remains a “beta” release.
- LibreOffice: is a project of the not-for-profit organisation, “The Document Foundation” <http://www.documentfoundation.org/>. Full information about the foundation can be found at that page.
- MathJax: is a joint project of the American Mathematical Society, Design Science, Inc., and the Society for Industrial and Applied Mathematics. These organisations also provide major funding. The list of MathJax sponsors can be found at <http://www.mathjax.org/sponsors/>
- \LaTeX -access: was created by Alastair Irving and Robin Williams. They ask that if you are interested in assisting with the project, even in a small way, to e-mail them. The last update to their sourceforge repository was on 27th May 2012.

Of course, some of the above appear to have substantial funding and community support. It is possible that their future is more secure than some of the commercial products we have used.

Specific risks and barriers

We attempt to capture some of the more specific remaining risks and barriers noted throughout the project. We have undoubtedly missed something and will probably update this section in the future.

Some methods are currently reliant on our own proof of concept work rounds. This is a **major risk**. Our proof of concept work rounds may not be reliable, may not be updated and ultimately may not be the correct approaches. They were **created to overcome known** bugs, issues or missing functionality in software.

- **Unclear situation with respect to native MathML support in browsers:** the lack of native browser support for MathML rendering led us to MathJax as an interim solution for large print/small screen devices while retaining the possibility of speech in IE with MathPlayer. We found that a full \LaTeX to MathML converter was needed for completely correct speech. However, this format cannot be rendered by most browsers. MathJax is not the work round — the particular manner in which we are producing files which render using MathJax is a work round.

We have provided a proof of concept renderer for PlasTeX which retains the \LaTeX of equation environments within the HTML and incorporates MathJax into the header. This renderer should be completed and submitted to PlasTeX. PlasTeX was a chosen technology due to the production of an interim document model which would allow rendering to multiple formats perhaps including mind-maps or other visually organised notes. It was also the simplest method to create a proof of concept of MathJax. In this respect we believe that improvements to PlasTeX offer significant gains.

- **Unclear situation with respect to linebreaking:** the `breqn` package cannot be used in master documents as no \LaTeX to MathML converter, including those we use (TeX4ht and MathJax) supports it. Automated linebreaking is not available in all output formats. The `breqn` package is the only available method to produce hard copy reflowed, large print mathematical documents. MathML3 supports linebreaking and MathJax (with the HTML-CSS renderer) supports this. However, MathJax webpages do not currently produce stable enough hard copy print output. TeX4ht and similar do not yet support MathML3. TeX4ht are currently undecided whether to support the `breqn` package or to use the automated linebreaking in MathML3 <http://www.cvr.cc/mathml-3-and-tex4ht/>.

The lack of support for any automated line-breaking in TeX4ht, along with the lack of browser support for MathML is what led us to MathJax as an interim solution for large print/small screen devices with some possibility of speech. However, TeX4ht remains the only software which produces a screenreader ready version. Automated linebreaking for mathematics is a fairly young technology; it is hoped that the correct method becomes clearer with time.

Knowledge that automated linebreaking might occur is required for authors to protect subparts of equations from linebreaking or to ensure that they have a full understanding of the implications of linebreaking for their encoding of content. In the present methods authors are divorced from direct experience of this when compiling their master as `breqn` cannot be used directly.

- **PlasTeX missing functionality:** code to deal correctly with `newtheorem` commands in \LaTeX appears to be incomplete; a basic and incomplete work round to this provided in our additional files does not encompass numbering or labels and references to any environments produced by `newtheorem`. The code is an incorrect approach put in place to avoid loss of the environment title. The missing functionality is reported at http://sourceforge.net/tracker/?group_id=120835 (bug 3061855, reported 2010).

It does not seem possible to insert alternative text for included images using PlasTeX. This is a key barrier to using this format with a screenreader regardless of the accessibility of the renderer used.

The renderers provided by PlasTeX are not aware of all equation environments. However, we do not recommend use of these in any case.

- **MathJax vs PlasTeX numbering:** We have encoded a requirement that equation numbering is by subsection and that the equation counter is reset by hand at the start of all sections and subsections and altered nowhere else (`tag` and `notag` also cannot be used but this is a stricter requirement as they can't be transformed by some software). This soft requirement on the numbering mode is very restrictive and applied solely to ensure that the numbering in all other formats is equivalent, but not even identical, to the PlasTeX produced HTML+MathJax rendering. The HTML+MathJax format is intended for large print/small screens and the output is broken into pages of a crosslinked website. This will greatly improve navigation for some students. Currently we "ask" MathJax to number the equations on each subpage.

Requiring PlasTeX to produce the numbers may be possible and this would be a more appropriate method. The current risk is that authors either forget to reset page numbers, choose not to or ask PlasTeX to break the document in a non-default manner. All formats will then have internally consistent but non-equivalent equation numbering.

- **Requirement to use local Unicode transformation files with TeX4ht:** In order to produce completely correct speech in the XHTML+MathML format, for every symbol permitted, unicode fonts had to be used. However, this produces unicode ligatures within the English text and these **cannot** be spoken correctly. This is a known behaviour <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=307647> and the recommended approach is to alter the unicode transformation files. This was completed for some but not all `unicode.4hf` files in the system. Rather than forcing a user/maintainer to alter parts of TeX4ht we have introduced local versions for the particular command we recommend. This adds an additional risk in that the transformation will appear visually correct if this file is not present in the directory but will not be read correctly by screenreader or text-to-speech.

In order to produce correct rendering of the various permitted symbols in the transformation to Word+MathType we must use the unicode expected by MathType (some are in the Unicode Private Use Area, see <http://www.dessci.com/en/support/mathtype/tech/encodings/mtcode.htm>). A second `unicode.4hf` file for the `oolatex` command was produced. The unicode produced by this command will not render successfully in the interim office formats, only once the full sequence of transformations to MathType format has been completed. There is a risk that interim formats will be used in the belief they are correct as this does not affect all symbols. There is a risk that if the `unicode.4hf` file is missing from the directory that the transformation to Word+MathType will be incorrect.

Considering both of the above factors, there is a risk that the two different `unicode.4hf` files, for XHTML and office transformations will be mistaken for each other. The files cannot be renamed. Due to this the two TeX4ht transformations **must** take place in different directories. This is an additional risk.

Despite the above, TeX4ht was retained as the most stable and configurable \LaTeX to MathML transformation experienced and the one in which additional \LaTeX packages are likely to be supported without production of configuration files (TeX4ht was the only transform to be successful on a test document using author provided documentclass and packages). This

is the only method by which correct speech has been produced for mathematics alphabets such as blackboard bold, calligraphic etc. in XHTML and MathML.

- **Requirement to use local configuration files with TeX4ht:** the local configuration files provided allow included images to be dealt with and alternative text for these to be inserted into the XHTML+MathML format. The configuration files are also necessary to combat difficulties with sub- and super-scripts explained at <http://www.tug.org/applications/tex4ht/mn3.html#QQ1-3-14>. Three possible work rounds are advised on that page but only the one we have used functions with the office transform as well as the XHTML+MathML transform. The main risk is that the configuration files are not present in the directory. This will cause unpredictable behaviour depending on the content of the \LaTeX file. The secondary risk is that users do not note that they must update the configuration file as per the content of their preamble.
- **LibreOffice bugs with MathML import:** This bug is reported at https://bugs.freedesktop.org/show_bug.cgi?id=47414. The bug also exists in OpenOffice and dates back to **at least** 2006. Our proof of concept work round for this produces a variant of the \LaTeX file which forces sub- and super-scripts to appear not to be a single symbol and all array cells to be non-empty (by effectively introducing "" in both cases). This forces the output from TeX4ht to produce MathML within the ODT format document which avoids the known cases of the bugs. LibreOffice is used as an interim format (after use of TeX4ht) to enable the ultimate production of Word documents for use with MathType. Multiple transformations of this type hold their own risk as errors may accumulate. It is possible that different software would be more appropriate.

Other risks and barriers include:

- \LaTeX is both more “relaxed” and provides far greater flexibility than the transformation methods. All of the following may result in failed or no output:
 - Common typos;
 - Common author practices which are not recommended but apparently tolerated by \LaTeX ;
 - Incorporating untested \LaTeX packages;

- Change in preamble ordering.

The guidance on writing \LaTeX documents must be followed precisely and documents checked carefully for the above if output fails. Further information on the above is given in the guidance document.

- Known bugs, issues or missing functionality for which there is no work round (tolerances):
 - The symbol, command and environment set permitted by all transforms is relatively limited. Some authors may have difficulty encoding their learning resources within the restrictions.
 - In \LaTeX certain types of environment can not be permitted within lists.
 - The method for working with EPS diagrams and psfrag may be a restriction which some authors are unwilling to work within.
 - Lists of theorems, figures, glossaries and indexes cannot be produced from \LaTeX .
 - The eqnarray environment is not numbered by MathJax <https://github.com/mathjax/MathJax/issues/229> though see above risks associated with the current approach to numbering in this format.
 - Mathematics alphabets (e.g. blackboard bold, calligraphic) appear not to be voiced correctly when published to MathPage by MathType and read by MathPlayer (and hence any speech technology reading the page). This is also true for MathJax pages rendered using MathPlayer but **not** the case for TeX4ht output.
 - Some two dimensional layouts simply cannot be accommodated beyond a certain font size. This requires the document author to consider elision but they may have difficulty ascertaining what types of structure or sizes of structure may cause reading difficulties. See \LaTeX guidance notes for comments on this.
 - URLs, in particular still sometimes overrun the RHS as they contain no natural breaking points and these cannot be introduced while retaining the live link in the HTML/XHTML versions.
 - URLs containing ampersands cannot be included in documents which are transformed to XHTML formats.
 - Commuting diagrams with diagonal arrows cannot be produced without working in a separate document, compiling it to a PDF and including the PDF as an image. All available packages were tested and all fail to be incorporated (or disrupt the transformations).

- Missing output formats:
 - A clear process for producing correct UK Braille did not emerge, all of the tested software was either too constrictive for incorporation in the general method, too unstable or too difficult to realistically evaluate in the time frame. Various possibilities for working with the \LaTeX source exist and were trialled. None have a symbol set as large as the general method but the focus on software which retains the \LaTeX commands which are not transformed is a possible work round. It is recommended that departments work closely with any student using Braille as a reading format in exploring these methods and brief information about them is in the transform guidance document.
 - The process for working with \LaTeX beamer documents not yet stable but we hope to deliver some guidance on this in the future as it is a commonly required set up.
 - A clear process for working with PowerPoints did not emerge although some guidance as to how to improve the accessibility of PowerPoint documents containing equations can be given and we hope to provide this in future.
 - Clear guidance on using handwriting as an input method did not emerge but some guidance on this can be given and we hope to provide this in future.
 - The production of a process for creating a visual organiser as an interface to long documents was not stable. It seems likely that this should be approached as a renderer from the PlasTeX document model but this was not considered at the time.

From a staff point of view the methods require a change in mindset away from typesetting or “What you see is what you get” (WYSIWYG) approaches to document production. This in itself may be a barrier.

Finally, a remaining barrier is that students and their support staff may not know how to use the software or resources effectively. For instance, using text-to-speech software with XHTML+MathML requires installation of MathPlayer and knowledge of how to request the text-to-speech software to read the resource. Guidance and possibly training for students and staff may be required.

5.1 Further development and sustainability

Staff producing mathematical resources for disabled students will embed the use of these methods in their own practice and in their advice to teaching staff. We did not approach students to provide feedback on the documents. In most cases they would have needed to learn new facets of an assistive technology and would have derived no immediate benefit from this time-consuming experience. As 'live' learning resources become available in multiple formats we will provide guidance for students and their support staff on how to read and use the formats effectively. Feedback from staff and students can then be used to develop the methods further. Dissemination of the methods may also result in further feedback and adjustments. As the methods are developed further or should new software functionality become available the online resources can be updated.

A second avenue for further development, which is required for sustainable use, is to work with the various software publishers and open source communities to patch bugs, improve functionality and ensure long term viability of the required software. Such work would eventually widen the viable user group of the proposed methods and improve the sustainability of them.

References

- [1] Cardetti, F., Khamsemanan, N. & Orgnero, M.C., "Insights regarding the Usefulness of Partial Notes in Mathematics Courses", *Journal of the Scholarship of Teaching and Learning*, vol. 10, no. 1, pp. 80-92, 2010.
- [2] Cliffe, E. & Rowlett, P. eds., "Good Practice on Inclusive Curricula in the Mathematical Sciences", National HE STEM Programme: Mathematical Sciences HE Curriculum Innovation Project, 2012. Available at: <http://mathstore.ac.uk/node/2095> [Accessed 27-May-2012].
- [3] Chapman, V., "Strategies for Creating Inclusive Programmes of Study (SCIPS)", 2005/7. Available at: <http://www.scips.worc.ac.uk/> [Accessed: 28-Oct-2011].

- [4] Loughborough University, "Disabilities: Academic Resource Tool (DART)", 2005. Available at: <http://dart.lboro.ac.uk/tool/index.html> [Accessed: 01-Nov-2011].
- [5] Maddox, S., "Mathematical equations in Braille", *MSOR Connections*, vol. 7, no. 2, pp.45-48, 2007.
- [6] Whapples, M., "Obtaining Braille mathematical documents", *MSOR Connections*, vol. 7, no. 3, pp. 18-21, Oct. 2007.
- [7] Stanley, P. "Assessing the Mathematics Related Communication Requirements of the Blind in Education and Career", in *ICCHP '08 Proceedings of the 11th international conference of Computers Helping People with Special Needs*, pp. 888891, 2008.
- [8] Cliffe, E. "Accessibility of mathematical resources: the technology gap", *MSOR Connections*, vol. 9, no. 4, pp. 3742, Nov. 2009.
- [9] Rowlett, E. J. "Accessibility in MSOR: one student's personal experience", *MSOR Connections*, vol. 8, no. 1, pp. 2730, Apr. 2008.
- [10] Rowlett E. & Rowlett, P., "Visual impairment in MSOR", *MSOR Connections*, vol. 9, no. 4, pp. 43-46, 2009.
- [11] RNIB, "Clear Print", 2011. Available at: http://www.rnib.org.uk/professionals/accessibleinformation/text/pages/clear_print.aspx [Accessed May 27, 2012].
- [12] Mole, J. & Peacock, D., "Learning, teaching and assessment: A guide to good practice for staff teaching d/Deaf students in science and engineering." University of Wolverhampton, 2005.
- [13] Easterbrooks, S., Stephenson, B. and Mertens, D., "Master Teachers' Responses to Twenty Literacy and Science/Mathematics Practices in Deaf Education," *American Annals of the Deaf*, vol. 151, no. 4, pp. 398-409, 2006.
- [14] G. Perkin and T. Croft, *The dyslexic student and mathematics in higher education*, *Dyslexia*, vol. 13, no. 3, pp. 193-210, Aug. 2007.
- [15] Graves, L. & Plant, S. "An Attitudinal Study of Pen Technology and Web-based Recordings to Accommodate Students with Disabilities in Post-Secondary Science, Technology, Engineering, and Math Courses," *The Impact of Tablet PCs and Pen-based Technology on Education: Going Mainstream*, 2010, pp. 39-46, 2010.

- [16] Webb, S. D., "Accessibility of University Mathematics", *MSOR Connections*, vol. 11, no. 1, pp. 4247, 2011.
- [17] Freda, C. et al., "Dyslexia: Study of Compensatory Software Which Aids the Mathematical Learning Process of Dyslexic Students at Secondary School and University", *Computers Helping People with Special Needs*, LNCS 5105/2008, pp. 742-746, 2008.
- [18] Iansyst, "Technology For Supporting Dyslexic Students Using Mathematical Notation And Scientific Language", 2007. Available from: <http://www.dyslexic.com/articlesearch.asp?type=Technology>. [Accessed: 31-May-2012].
- [19] Sojka, P. & Růžička, M., "Single-source publishing in multiple formats for different output devices", *TUGboat* vol. 29, no. 1, XVII European TeX Conference, pp. 125-131, 2007.
- [20] Godfrey, A. J. R., "Putting it all together — A blind person's perspective on developing a toolbox for document preparation", *Inftyreader.org*, 2012. Available from: <http://inftyreader.org/18GodfreyPutting.pdf>, [Accessed 15-May-2012]
- [21] Sojka P. & Plch, R., "Technological challenges of teaching mathematics in a blended learning environment", *International Journal of Continuing Engineering Education and Life Long Learning*, vol. 18, no. 5, pp. 657665, Jan. 2008.
- [22] Cooper, M., "Making Mathematics Teaching Inclusive – Access for Disabled Students to Symbolic Languages in Electronic Media", *MSOR Connections*, vol. 6, no. 4, pp. 29-37, 2006.
- [23] Cooper, M., Lowe, T. & Taylor, M. "Access to mathematics in web resources for people with a visual impairment: Considerations and developments in an open and distance learning context", *Computers Helping People with Special Needs*, vol. 5105/2008, pp. 926933, 2008.
- [24] W3C, "Mathematical Markup Language (MathML) Version 3.0", 2010. Available at: <http://www.w3.org/TR/MathML3/>. [Accessed 31-May-2012]
- [25] Stanley, P. B., Bledsoe, C. & Karshmer, A. I., "Utilizing Scalable Vector Graphics in the Instruction of Mathematics to the Print Impaired Student", in *Computers Helping People with Special Needs*, 2004, pp. 626629.
- [26] Downes, M. "Breaking equations", *TUGboat*, vol. 18, no. 3, 1997.

- [27] W3C, “The W3C MathML software list”, 2012. Available at: <http://www.w3.org/Math/Software/>. [Accessed: 31-May-2012].
- [28] McKain, D., Talk:“Translation of TeX to XML and MathML” given at Technical Aspects of Mathematical Content workshop, Open University, 9th Sept. 2009. Video and slides available via: <http://stadium.open.ac.uk/stadia/preview.php?whichevent=1393>, http://stadium.open.ac.uk/webcast-ou/documents/mc09_david_mckain_conversion.pdf.
- [29] Stamerjohanns, H. et al., “MathML-aware article conversion from LaTeX. A comparison study”, Towards a Digital Mathematics Library, Grand Bend, Ontario, Canada, July 2009, pp.109-120, 2009.
- [30] Gurari, E. “TeX4ht: HTML Production”, TUGboat, vol. 25, no. 1, 2004.
- [31] Arnold, T., “plasTeX: Converting LaTeX Documents to Other Markup Languages”, The PracTeX Journal, TeX Users Group, vol. 3, 2008.
- [32] Stamerjohanns, H. et al., “Transforming Large Collections of Scientific Publications to XML”, Mathematics in Computer Science, vol. 3, pp. 299307, Feb. 2010.
- [33] Ginev, D. et al., “The LaTeXML Daemon: Editable Math on the Collaborative Web”, in Intelligent Computer Mathematics, vol. 6824, J. H. Davenport, W. M. Farmer, J. Urban, and F. Rabe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 292294.
- [34] Cervone, D., “MathJax: A Platform for Mathematics on the Web.” Notices of the AMS, vol. 59, no. 2, pp. 312-316, 2012.
- [35] Soiffer, N. “MathPlayer: web-based math accessibility”, in Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility, New York, NY, USA, 2005, pp. 204205.
- [36] Gillan, D. J. et al., “Cognitive Analysis of Equation Reading: Application to the Development of the Math Genie”, in Computers Helping People with Special Needs, 2004, pp. 630637.
- [37] Hennings, W., “Converters from PC Textprocessors to LaTeX — Overview”, 2011. Available at: <http://tug.org/utilities/texconv/pctotex.html>. [Accessed: 31-May-2012].

- [38] Hennings, W. “Converters from LaTeX to PC Textprocessors — Overview”, 2011. Available at: <http://tug.org/utilities/texconv/textopc.html>. [Accessed: 31-May-2012].
- [39] AMS, “TeX Resources”, 2012. Available at: <http://www.ams.org/publications/authors/tex/tex>. [Accessed: 31-May-2012]
- [40] CTAN, “The Comprehensive TeX Archive Network” at <http://www.ctan.org/>. [Accessed 31-May-2012]
- [41] Gregorio, E. “Horrors in LaTeX: How to misuse LaTeX and make a copy editor unhappy”, TUGboat, vol. 26, no. 3, pp. 273279, 2005.

A List of technologies used during the project

Please be aware that technologies were **discarded** during the evaluation process if they were too constrictive or unstable for our purposes or if the results difficult to realistically evaluate. This is not to suggest that these technologies could not fulfill useful roles in enabling access to mathematics or even that we will not return to them in future work. Hence, this list contains technologies that **cannot** currently be used within our methods as well as those that can.

- TeXLive <http://www.imagemagick.org>;
- \LaTeX : We encountered a wide collection of packages incidentally; those tested for a specific purpose are listed below:
 - AMS- \LaTeX : <http://www.ctan.org/pkg/amslatex>;
 - beamer: <http://www.ctan.org/pkg/beamer>;
 - beamerarticle: <http://www.tex.ac.uk/CTAN/macros/latex/contrib/beamer/base/beamerarticle.sty>;
 - extsizes: <http://www.ctan.org/pkg/extsizes>;
 - breqn: <http://www.ctan.org/pkg/breqn>;
 - geometry: <http://www.ctan.org/pkg/geometry>;
 - babel: <http://www.ctan.org/pkg/babel>;
 - fontenc: <http://www.ctan.org/pkg/fontenc>;

- verbatim: <http://www.ctan.org/pkg/verbatim>;
- spverbatim: <http://www.ctan.org/pkg/spverbatim>;
- listings: <http://www.ctan.org/pkg/listings>;
- url: <http://www.ctan.org/pkg/url>;
- latexsym: <http://www.mackichan.com/index.html?techtalk/500.htm>;
- bm: <http://ctan.org/pkg/bm>.
- Table environments:
 - * tabu: <http://www.ctan.org/pkg/tabu>;
 - * longtable: <http://www.ctan.org/pkg/longtable>;
 - * tabulary: <http://www.ctan.org/pkg/tabulary>;
 - * tabularx: <http://www.ctan.org/pkg/tabularx>.
- ntheorem: <http://www.ctan.org/pkg/ntheorem>;
- thmtools: <http://www.ctan.org/pkg/thmtools>.
- Images:
 - * Graphicx: <http://www.ctan.org/pkg/graphicx>;
 - * epslatex: <http://www.ctan.org/pkg/epslatex>;
 - * psfrag: <http://www.ctan.org/pkg/psfrag>;
 - * epstopdf-pkg: <http://www.ctan.org/pkg/epstopdf-pkg>;
 - * amscd: <http://www.ctan.org/pkg/amscd>;
 - * xymatrix: <http://www.ctan.org/pkg/xymatrix>;
 - * pb-diagram: <http://www.ctan.org/pkg/pb-diagram>;
 - * dcpic: <http://www.ctan.org/pkg/dcpic>.
- Matching brackets to locate \LaTeX typos:
 - match_parens: http://www.ctan.org/pkg/match_parens;
 - check-parens: <http://www.ctan.org/pkg/check-parens>.
- Converting image formats:
 - ImageMagick: <http://www.imagemagick.org>.
- Visual \LaTeX display and editing:
 - LyX: <http://www.lyx.org/>;
 - BaKoMa: <http://www.bakoma-tex.com/>.

- \LaTeX converters:
 - TeX4ht: <http://www.tug.org/tex4ht/>;
 - PlasTeX: <http://plastex.sourceforge.net/>;
 - LaTeXML: <http://dlmf.nist.gov/LaTeXML/>;
 - Tralics: <http://www-sop.inria.fr/marelle/tralics/>;
 - MathJax: <http://www.mathjax.org/>;
 - LaTeXMathML: <http://www.maths.nottingham.ac.uk/personal/drw/lm.html>;
 - ASCIIMathML: <http://www1.chapman.edu/~jipsen/mathml/asciimath.html>.
- Converting to Word:
 - TeX4ht: <http://www.tug.org/tex4ht/>;
 - LibreOffice: <http://www.libreoffice.org/>;
 - MathType: <http://www.dessci.com/en/products/mathtype/>.
- Converting from Word:
 - Word2TeX: <http://www.chikrii.com/products/word2tex/>;
 - Word-to- \LaTeX : <http://www.wordtolatex.com/>;
 - GrindEq: <http://www.grindeq.com/>.
- To produce speech:
 - MathPlayer: <http://www.dessci.com/en/products/mathplayer/>;
 - MathGenie: <http://www.logicalsoft.net/Math.html>, <http://rskipper.myweb.usf.edu/TMVIProject.html>.
- Used in attempts to produce various Braille formats:
 - Scientific NoteBook: <http://www.mackichan.com/index.html?products/snb.html>;
 - Duxbury: <http://www.duxburysystems.com/dbt.asp>;
 - ChattyInfty: <http://www.sciaccess.net/en/ChattyInfty/>;
 - Lambda: <http://www.lambdaproject.org/>;
 - \LaTeX -access: <http://latex-access.sourceforge.net/>;

- BrITeX: <http://brltex.sourceforge.net/>;
- LaTeXLex: <http://mesarhameed.info/projects/latexlex>.
- Used in attempts to improve accessibility of PowerPoint documents:
 - MicroSoft Word and PowerPoint;
 - LecShare: <http://lecshare.com/>;
 - PowerTalk: <http://fullmeasure.co.uk/powertalk/>.
- Used in attempts to produce visual interfaces to document parts:
 - FreeMind: <http://freemind.sourceforge.net/wiki/index.php/>;
 - FreePlane: <http://freeplane.sourceforge.net/wiki/index.php/>.

For evaluating the resources, in addition to some of the above, we used:

- A range of modern Internet browsers and PDF viewers;
- The Stix fonts: <http://www.stixfonts.org/>;
- Kindle and Sony e-book readers;
- JAWS: <http://www.freedomscientific.com/jaws-hq.asp>;
- TextHelp: <http://www.texthelp.com/UK/our-products/readwrite>;
- A text editor which JAWS could interact with — I used programmers notepad <http://www.pnotepad.org/>.

For producing proof of concept work rounds we used:

- Flex: <http://flex.sourceforge.net/>;
- C and Python.